

シーケンス図とクラス図の生成に適したユースケース記述形式

A proposed for new use case description to generate sequence and class diagrams

奥村 和恵[†] 金澤 典子[†] 塚本 享治[†]

Kazue Okumura Noriko Kanazawa Tsukamoto Michiharu

1. はじめに

システム開発で UML などの図を中心に開発する手法が、広く使われている。しかし UML 図の作成は慣れない開発者にとって難しい。我々はより簡単に図を作成するために、仕様書の一つであるユースケース記述を、シーケンス図とクラス図へ変換するシステムの開発を進めている。

このシステムではユースケース記述中の、システムの基本的設計情報とシナリオから、UML 図へ変換する。通常のユースケース記述[1]からは、モデルに必要な要素を正確に抽出できなかった。本稿では変換に適したユースケース記述の形式を提案し、変換ツールを試作して実験を行った。この実験結果について報告する。

2. これまでの研究

[1]では日本語を制限したユースケース記述から、UML 図の自動生成を試みた。課題は変換元の文章を制限に沿って、厳密に書かなければならないことだった。制限を無視すると、正確な図が生成出来なかった。原因の一部に、同じオブジェクトを異なる表現で記述すると、全く違うオブジェクトと認識されるという問題があった。

これを解決するため、ユースケース記述の制限を緩和しても、正確に変換できるシステムの作成を試みた。解決方法はユースケース記述内に書かれるシステムの基礎情報を、図への変換に利用した。

本実験で使った変換ツールは、[1]と同じ図 1 の方法である。まず第4章で提案するユースケース記述文書を用意し、多機能日本語処理ライブラリ[3]で構文解析を行った。そして解析結果から図に必要な要素を抽出し、XMI 形式へ変換した。最後に UML モデリングツール[4]に読み込ませ、シーケンス図とクラス図を生成した。

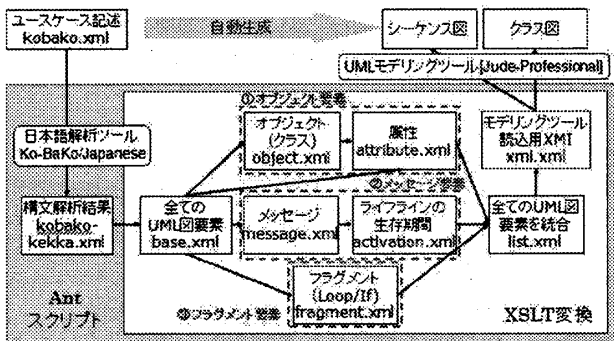


図 1 変換の全体像

3. シナリオの文構造の分析

ユースケース記述はシステムの動作内容の情報が中心である。後述するユースケース記述内の他の情報と区別するため、本稿では動作内容の記述部分を「シナリオ」と呼ぶ。

[†] 東京工科大学大学院バイオ・情報メディア研究科
メディアサイエンス専攻

このシナリオを構文解析[3]すると、単語の助詞と文の構造が出力される。ツールではこの 2 点により、各文から UML 図の要素を抽出した。

単語がシーケンス図のどの要素にあたるかは、助詞から判断できる予想した。助詞と図要素の組み合わせにはパターンがある。このパターンを適用することで、文中の単語と図要素の対応が分かると推測した。

シナリオに多く見られる助詞と図要素のパターンを分析するため、[5]にある 5 ケのユースケース記述 (計 71 ステップ) を用意した。これを構文解析[3]し、解析結果を分析したのが表 1 である。(頻出パターンは背景色付き)

表 1 シナリオ文の[3]による助詞解析結果のパターン

頻度	送信側 Object	受信側 Object	引数 1	引数 2
3%	GA (が) 66%	KARA (から)	NI (に)	WO (を)
3%		HE (へ)	WO (を)	-
8%		TAISURU (対する)	WO (を)	-
1%		NI (に) 23%	TAISURU (対する)	WO (を)
18%			WO (を)	-
4%		WO (を) 9%	WO (を)	-
3%			-	-
6%		-	TAISURU (対する)	-
3%		- 20%	WO (を)	-
17%			-	-
8%	HA (は) 24%	KARA (から) 14%	WO (を)	-
6%			TAISURU (対する) TOSHITE (として)	WO (を)
8%		WO (を)/NI (に)	-	-
1%		GA (が)	NonDA (か否か)	-
10%	その他	その他	&/NI/WO/-	WO/-

※頻度=全体を 100%とする

分析結果から、シーケンス図の各要素に付く助詞には傾向があると判断できた。この傾向により、単語がどの図の要素か助詞で判別できる。

実験ではツールが自動的に図要素を判別した。判別基準が助詞の「頻度」だけでは、誤解析が多かった。そこで助詞の「優先順位」という判別基準を加えて、精度を上げようと試みた。

(1) 助詞の頻度

出現頻度が多い順に、助詞を場合分けした。送信側オブジェクトの助詞の変化に伴い、受信側オブジェクトの助詞も変化した。最頻出パターンは「送信側オブジェクト="GA (が)"、受信オブジェクト="NI (に)"、引数="WO (を)"」の組合せであった。

(2) 助詞の優先順

"KARA (から)"や"HE (へ)"などの助詞は出現頻度は低くても、受信側オブジェクトにしか使われない。図要素の抽出は、受信側オブジェクトを引数より優先して抽出する。そのため以下のルールが当てはめられる。

- ① 受信側 Object=[KARA | HE] → 引数=[NI | WO]
 ② 受信側 Object=[NI | WO] → 引数=[WO | -]

4. 変換に適したユースケース記述形式の提案

通常のユースケース記述には契約書という側面がある。契約書として必要でも、設計書として必要ない情報が載っていたり、設計に必要な情報が載っていないことがある。そこで変換ツールで使うために、必要な情報だけを集めた新しいユースケース記述の形式(表2)を提案する。

表2 提案するユースケース記述形式

ユースケース名	ユースケース記述が実現するユースケース名。関連クラスの鍵となる。
アクタ	シナリオに登場するアクタ、登場人物。システムの実行者。
システム	シナリオに登場する、ユースケースやコントローラの実行者。データベースに保持するデータを持つ。関連クラスの鍵となる。
コントローラ	ユースケースを実現するために行う処理、イベント。一時的なデータしか持たず、データベースにデータを保持しない。
リソース	シナリオに登場する情報源(データ)。データベースに登録するデータを持つオブジェクト、エンティティ。
インターフェイス	シナリオに必要な画面。ロバストネス図のバウンダリオブジェクトにあたる。
オブジェクトの属性	アクタやエンティティが持つ属性。データベースの属性列にあたる。
シナリオ	各オブジェクトの動作を時系列で表す。設計レベルでオブジェクトに振舞を与える。

ユースケース記述にはコントローラやインターフェイスなど、変換対象の図には直接必要ない情報も含めた。現在開発中のツールを今後、ロバストネス図などに対応させるためである。

なお表2の情報は、システム分析フェーズで既に明らかになっていると仮定した。([2]の開発手法より)

5. 実験と考察

最初に4章の形式に従った、変換専用のユースケース記述(表3)を用意した。これを図1のツールに読み込ませ、変換実験を行った。

表3 変換ツールへ読み込むユースケース記述

ユースケース名	<usecase>配付を依頼する</usecase>
アクタ	<actor>営業</actor>*
システム	<system>拡販資料配付システム</system>*
コントロール	<control>在庫数を取得する</control>*
リソース	<entity>拡販資料</entity>*
インターフェイス	<bound>配付先入力画面</bound>*
オブジェクトの属性	<attribute class="配付依頼">管理ID</attribute>*
シナリオ	<step><構文解析>1. 営業は以下の情報を入力し、配付先を照会する</構文解析>*** </step>

※出典:[5] p.82「配布を依頼する」より

実験の結果、[1]では変換できなかったユースケース記述が、本ツールでは解析・変換できた。理由は3章の助詞の解析を強化したためと予想される。しかし生成されたUML図の正確性に問題があった。特にクラス図に必要な、「クラス間の関連性」が抽出できなかった。

4章でユースケース記述の形式を指定した影響を確かめるため、抽出したオブジェクト要素について本ツールと[1]を比較(表4)した。比較項目①の抽出率とは、全オブジェクト数のうち、ツールが抽出できたオブジェクトの割合である。比較項目②の正確性とは、ツールが抽出したオブジェクト数のうち、正しいオブジェクト要素を抽出できた確率である。

表4 ツールのオブジェクト要素の抽出率と正確性

	①抽出率	②正確性	サンプル数
本ツール	96%	59%	8
[1]ツール	93%	45%	4

本ツールは[1]と比べて、3%の抽出率と14%の正確性の改善が見られた。抽出率の3%向上は誤差の範囲内であるため、改善したとは言いがたい。正確性の14%向上については、ユースケース記述形式を拡張した影響があったと言えるだろう。

今回の実験はサンプル数が少ないため、表4の結果から確かに改善されたとは断定できない。正確性が59%と低いことが問題である。

6. おわりに

ユースケース記述をシーケンス図とクラス図へ変換する実験を行った。変換元文書には第4章で形式を拡張したユースケース記述を用いた。

実験の結果、助詞の解析を強化したことにより、[1]では解析できなかった文書を解析できた。またユースケース記述の形式を拡張したことで、オブジェクト要素の抽出精度が改善した。

この実験で最終的に、ユースケース記述に掛けていた制限を緩和できた。課題は、抽出した図要素の正確性が低い(59%)こと、生成されたUML図が正確に仕様を反映していないことである。

参考文献

- [1] 奥村和恵, “日本語構文解析を用いたユースケース記述からシーケンス図への変換”, 情報処理学会研究報告 2008-SE-159(1), 2008
- [2] ダグ・ローゼンバーク, マット・ステファン, “ユースケース駆動型開発実践ガイド”, 翔泳社, 2007
- [3] “Ko-BaKo/Japanese”, (株)日本システムアプリケーション, <http://www.jsa.co.jp/LANG/ko-bako/>
- [4] “Jude Professional”, (株)チェンジビジョン, <http://jude.change-vision.com/jude-web/index.html>
- [5] 浅井麻衣, 重田正俊, 橋本大輔, 浜口弘志, 藤井啓詞, “現場のUML”, ソーテック社, 2006